목차

1. 과제 배경

기후변화는 우리 삶에 점점 영향을 끼치고 경제적,인명 피해도 발생하고 있다. 21년 6월 캐나다에서 기온이 50도 가까이 치솟아 100여명이 숨졌고, 21년 7월 중국에서도 폭우로 200명 이상의 사망자가 발생하였다. 스위스리(swiss Re)는 지구 평균 온도 2.6도 상승시 우리나라의 국내총생산(GDP)이 9.7% 약 188조원의 피해가 발생할 것이라고 예측했다. 이를 해결하기 위한 노력중 하나이고 이번 대선 토론에 나와서 이슈가 되었던 RE100캠페인이 등장하였다.

RE100이란 Renewable Electricity 100의 약자로 기업에서 사용하는 전력의 100%를 재생에너지로 대체하자는 국제적 기업간 협약 프로젝트이다. 재생에너지는 석유화석연료를 대체하는 태양열, 태양광, 바이오, 풍력, 수력, 지열 등에서 발생하는 에너지를 말한다. 기업활동에 필수적으로 필요한 전기를 온실가스를 배출하지 않는 재생에너지로 생산된 전기로 사용하겠다는 목적이 있다.

우리나라는 롯데칠성음료, KB금융그룹, LG에너지솔루션, 현대자동차 등 19개 기업이 동참하고 있으며 세계적으로는 3M, Apple, Airbnb등 370개 기업이 참여하고 있다. 세계적으로 온실가스를 대량으로 배출하는 기업의 사회적 책임을 묻기 시작하고, RE100 회원사 중 일부 기업들이 협력업체에게도 재생에너지 전기를 사용하여 납품하도록 요구하는일이 생기면서 RE100은 기후위기 대응을 넘어 국내 주요 기업의 수출경쟁력에 직결되는 요소가 되었다. 이제 RE100은 선택이 아니라 필수라고 말할 수 있다.

이런 글로벌 이슈인 RE100의 이행수단으로서 유럽에서는 녹색요금제가 중심으로 사용되며 북미,아시아는 REC(신재생 에너지 공급 인증서)구매를 중심으로 이행하고 있다. 우리나라또한 올해 1월부터 REC의 거래가 시작되었다. 신재생 에너지 사업자가 신재생 에너지로 전력을 생산하면 그에 맞는 REC코인을 받을 수 있고 기업들은 그 REC코인을 구매해서 RE100을 수행했음을 인정받을 수 있다. 하지만 현재 REC거래는 다음과 같은 문제점이 있다.

현재 REC거래 체계는 매우 비효율적이다.

REC계약을 위해서 한국에너지공단은 공급사업자 선정입찰을 통해 낙찰된 사업자 정보를 공급의무자별로 정리하여 통보해야하는데 이는 절차적 번거로움, 낙찰자 정보의 수집 및 재작성, 송부 시 오류가능성이 존재한다. 또한 공급의무자는 공급사업자와 오프라인 서면 계약을 체결해야 해서 인력, 시간의 불필요한 소모가 필요하다. 그리고 REC설비확인 신처과 REC발급은 한국에너지공단 신재생에너지센터에서 하지만 REC거래는 전력거래소를 통해 이뤄져 공급의무자가 REC대급지급 시 각 데이터들이 정확한지 대조절차를 거쳐야 한다. REC의 발급과 매매단계의 주체가 상이함으로 REC거래에 대한 정보가 단절되 발생하는 문제이다.

현재 REC거래 시스템은 보안에 취약하다.

현 시스템은 중앙집중화의 시스템으로 구성되어 있다. 데이터베이스 또한 각자의 기관에서 필요한 정보를 각자가 소유하고 있다.(이는 위 비효율적 체계의 예시이기도 하다) 각자의 기관에서 유지중인 서버가 공격을 당할시에 업무가 중지될 위험이 존재한다. 그리고 각자의 데이터베이스를 가지고 있어 시간의 경과에 따라 데이터양이 늘어나는것도 문제가 된

다.

현재 REC거래 구조에도 문제가 있다.

현재 REC계약자체가 대용량 패키지 구조로 되어 있어, 전력수요자의 사용목적과 용량에 맞게 쪼개어 구매하기가 힘들다. 비싼비용을 지불하고 구매해야 하고 수수료를 요구하기 때문에 작은 규모의 사업자들은 REC구매에 접근하기가 어렵다. 또한 REC시장은 크게 계약시장화 현물시장 2가지가 있는데 현물시장의 경우 매도주문을 접수하고 해당 매도물량에 대해 높은가격을 제시한 주문자가 낙찰받는 구조이다. 태양광업계는 구조가 의무공급자에게만 권한이 크게 작용되고 있고, 이를 해결하기 위해 주식시장처럼 실시간 거래로 바꿔야 한다고 주장한다.

2. 과제 목표

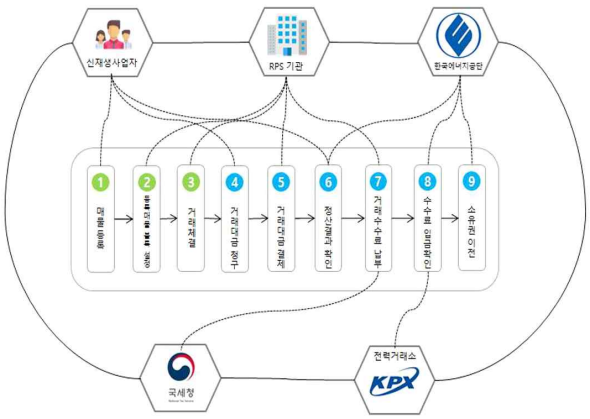


그림 1. Transaction procedure of spot market

그림1을 살펴보면 현재 REC거래 현물시장의 모습을 보여준다. 사업자가 REC를 등록하면 전력거래소에서 매물로 설정하고 공급 의무자는 REC를 돈을 주고 구매하고 대금 청구 및 대금이 오고 간다. 국세청에 대금 수수료가 납부되고, 과정이 완료되면 REC의 소유권이 이전된다.

이때 각 기관만이 소유하고 있는 데이터들이 있어 서로 대조가 필요하며 거래 과정도 복잡하다. 이 때문에 REC매매절차가 약 9단계로 복잡하여, 고령층 및 농민 사업자들의 불만들이 많았다. 그리고 각 기관의 데이터베이스의 위조, 해킹 등의 위협도 존재한다. 이럴 경우 직접적으로는 거래내용의 해킹으로 금전적인 피해가 발생할 수 있고, 간접적으로는 이런 공격이 감지되었을 경우에 REC거래 자체가 일시중단 되면서 피해가 발생할 수 있다. 또 기관만이 소유하고 있는 데이터들에 대해 발급내역과 대금정보가 불일치할 가능성이 있어 거래 소모비용이 증가한다. 또 이런 절차적 문제들로 인해 지정된 장소에 직접 방문할 필요가 있었고 서면계약 체결이 필요하다.

이런 문제들은 결국 각 데이터들이 서로 다른 기관에 분포해 있어서 생기는 문제이다.
이 문제는 하이퍼래저 패브릭을 사용해서 해결이 가능하다.

하이퍼래저 패브릭은 프라이빗 블록체인의 일종인 허가형 블록체인이다. 블록체인은 P2P 방식을 기반으로 하여 소규모 데이터들이 체인 형태로 연결되어 형성된 블록이라는 분산데이터 저장 환경에 데이터를 저장함으로써 임의로 수정할 수 없고 누구나 변경의 결과를 열람할 수 있게끔 한 기술이다. 블록에 모든 거래 내역이 기록되어 있고 모든 사용자들이 열람 가능하므로 위변조를 할 수 없다. 허가형 블록체인은 이 블록체인에 사용자를 허가받은 사용자에 한정한다. 이 REC거래는 기관과 판매자, 구매자에 한정되기 때문에 허가형 블록체인인 하이퍼래저 패브릭 사용이 필요하다.

이 기술을 사용하면 기관을 통해 거래를 요청하는 것이 아니게 된다. 등록된 기관이면 어느 기관이든 상관없이 접근 가능하며, 연관되어있는 각 기관들의 전체적인 인증 요청을 통해 거래의 타당성을 확인하여 결과가 일치해야 승인 값을 반환한다. 이 방식은 하나의 기관에서만 인증을 받는 것이 아니므로 보안에 매우 뛰어나다. 그리고 모든 기관이 공통된 정보를 소유할 수 있음으로 거래 장부의 투명성이 보장되어 있고 서로의 데이터를 대조할 필요가 없다. 그리고 고객들이 직접 그들만의 거래를 지향하므로 복잡한 거래 절차가 축소될 수 있다. 따라서 서면 계약이 불필요해지고 수수료 절감 효과도 노릴 수 있어 소규모 사업자들의 접근성도 좋아진다. 한국남부발전에서 작성한 보고서에 따르면 이를 통해 국민은 약 2억 6천만원(년), 기관은 약 5억 3천만원(년)의 사회적비용을 감소시킬 수 있을것이라고 한다.

따라서 탈중앙화 방식의 블록체인형 REC거래 중계모듈을 개발해 편의성 및 보안성을 높이고자 한다.

3. 요구 조건 분석

3.1 하이퍼래저 패브릭을 적용한 REC거래

➢ 현재 시스템상으로는 매매를 하는 단계마다 실행을 하는 주체가 다르며, 각 주체가 각각 데이터를 가지고 있어 단계가 복잡해지고 데이터 교차검증을 해야 하는 문제가 발생한다. 또한 해킹과 변조의 위험도 존재한다. 사회적 비용의 증가로도 이어진다. 따라서 REC에 블록체인을 적용할 필요성이 생긴다. 그렇다면 어떤 블록체인을 사용해야 할지 정해야 한다. 퍼블릿 블록체인과 프라이빗 블록체인이 있는데 간단한 비교표는 다음과 같다.

	퍼블릭 블록체인	프라이빗 블록체인
읽기 권한	누구나 열람 가능	허가된 기관만 열람 가능
거래 검증 및 승인	누구나 네트워크에 참여하면 거래 검증 및 승인을 수행	승인된 기관과 감독 기관
트랜잭션 생성자	누구나 트랜잭션을 생성	법적 책임을 지는 기관만 참여
합의알고리즘	부분 분기를 허용하는 자겁 증명이나 지분증명 알고리즘	부분분기를 허용하지 않는 BFT계열의 합의 알고리즘
권한 관리	모두가 모든 일 가능	private Channel, tiered system등을 통해 읽기 쓰기 권한 관리가 가능
예시	비트코인, 이더리움	IBM Fabric, Hyperledger fabric

표 2 퍼블릿, 프라이빗 블록체인 비교

REC거래는 공급의무자와 신재생사업자, 한국에너지공단, 전력거래소, 국세청등이 주체가

되는데 이런 신뢰성이 보장된 기관만이 사용할 수 있어야 한다.사전승인된 사용자만이 참여할 수 있어야 함으로 프라이빗 블록체인을 사용하고, 그중에서 최근 가장 대표적으로 사용되는 하이퍼래저 패브릭(Hyperledger Fabric)을 사용한다.
하이퍼래저 패브릭은 블록체인 프레임워크로 리눅스 환경에서 호스팅하는 프로젝트이다. 오픈소스로 공개되어있고 체인코드(chain code) 라는 스마트 계약을 호스팅한다. 체인코드를 사용하면 계약이 만들어졌을 때 같은 체인코드가 설치된 피어들 사이에서 정보를 서로 검증하고 저장하므로 강력한 보안을 얻을 수 있다.
해당 로직을 단순화하면 다음과 같다.

1.클라이언트 어플리케이션에서 SDK를 통해 트랜잭션 제안을 발생
2.체인코드의 보증정책에 명시된 노드들이 체인코드를 실행
3.각 결과값을 클라이언트 어플리케이션에 전달
4.결과값이 보증 정책을 만족시, 결과값은 오더링 서비스에 전달
5.오더링 서비스 블록의 순서를 결정해 채널의 모든 피어에 전달
6.모든 피어는 블록이 보증정책을 만족하는지, 장부가 트랜잭션이 일어나는동안 바뀌지 않았는지 확인
7.피어들은 블록을 채널의 체인에 덧붙이고 장부의 상태 업데이트

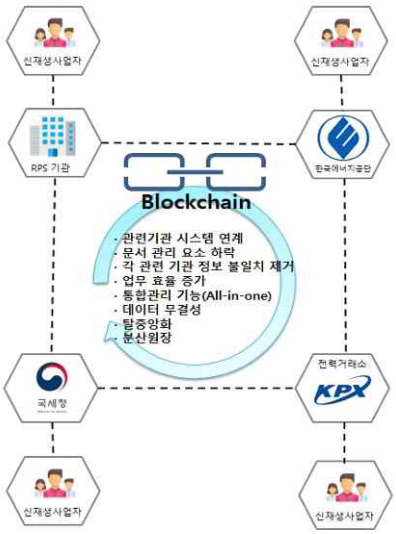


그림2 .REC transaction of consortium blockchain

블록체인을 적용하게 되면 그림1과 달리 모든 기관들이 같은 데이터를 사용할 수 있고 보안성도 훌륭한 서비스를 만들 수 있다.
따라서 Hyperledger Fabric Go contract API를 사용해서 블록체인 구현 및 REC거래모듈 개발을 하도록 한다.

3.2 서비스 제공

➢ 웹페이지를 이용하여 REC를 안전하게 거래할 수 있도록 한다.

단 여기서 사용하는 블록체인은 개방형 블록체인(public blockchain)이 아닌 허가형 블록체인(permissioned blockchain)이므로 신원 관리(로그인)가 필요하다.

웹서비스는 계약시장, 현물시장 두가지 시장으로 구분된다. 계약시장은 15~20년정도의 계약을 맺으며 REC의 가격을 고정하여 계약하는 방식이다. 현물시장은 현재 생산된 전력에 대한 REC를 거래하는 시장이다. 두가지 시장에 대한 판매, 구매와 정산이 가능하도록 한다. 그리고 현재의 REC거래의 단점이 복잡한 과정(매물등록->매물설정->거래체결->거래대금청구->거래대금결제->정산결과확인->거래수수료 납부->수수료입금확인->소유권이전)을 단순화(REC등록->거래대금 청구->입금확인->정산) 하여 제공한다. 구매과정과 세금결제관련내용도 단순화 하도록 한다.

4. 현실적 제약 사항 및 대책

거래하는 에너지가 실제로 신재생 에너지인지 검사할 방법이 없다. 이를 해결하기 위해서는 차후에 정부 부처와의 협약을 통해서 실제 신재생 에너지인지 검사해야 할 것이다. 신재생에너지 공급자에 대한 직접확인 과정은 단축될 수 없다.

블록체인에서 이미 올라간 계약은 수정과 삭제가 불가능하다. 그래서 만약 계약을 파기하고 다시 계약을 한다고 하면 이전 계약을 수정하거나 삭제할 수 없고 새로운 계약에 대한 블록체인을 다시 만들어서 올릴 수 밖에 없다. 이때 이전 계약의 블록체인 때문에 불필요한 저장공간이 필요하고 다시 블록체인을 만드는 과정에서 overhead가 발생할 수 있다. 수정과 삭제가 불가능하다는 것은 블록체인의 근본적인 속성 중 하나이기 때문에 고칠 수는 없고, 계약을 여러 번 할 때 생기는 overhead라도 줄이기 위해서 블록체인을 효율적으로 구성해야 할 것이다.

비트코인은 블록체인 기술을 사용한 대표적인 코인 중 하나이다. 비트코인 블록이 여러 개 쌓이면서 200GB 이상의 용량이 필요로 하게 되었다. 이처럼 REC 거래에서 계약이 여러 번 쌓여서 여러 블록체인을 생성하게 됐다면, 블록체인이 차지하는 저장공간이 많아질 수 있다. 그래서 만약 블록체인의 크기가 개인의 저장공간보다 더 커지게 된다면 노드를 잃어버릴 가능성이 있다. 이에 대한 해결방안으로는 사용자의 저장공간을 충분히 확보하거나 블록체인이 일정 용량이 쌓이면 처음부터 다시 블록을 쌓아가는 식으로 해결할 수 있을 것이다. 하지만 이는 임시방편이므로 제대로 된 해결방법을 생각해봐야 할 것이다.

5. 설계 문서

5.1 개발 환경

- 개발 언어

언어	역할
golang	블록체인 구동
html	웹 페이지 구현
css	html을 보조하여 웹페이지 구현
typescript	html, css와 함께 웹페이지 구현에 사용. javascript와는 다르게 구현할 때 type을 명시하므로 compile 단계에서 오류를 잡아낼 수 있다는 장점이 있다.

- 개발 도구

도구	역할
vscode	대표 IDE로 팀원 모두 수년간 vscode를 사용해왔기에 vscode를 사용하기로 함
git	대표적인 협업 도구로 code 백업에 사용
sourcetree	git 사용에 있어 UI를 접목해서 사용하기 편하게 해주는 툴
ubuntu	fabric 구현 시 대부분 linux를 사용하기에 ubuntu를 사용하기로 하였다.
docker	linux 환경에서 컨테이너 환경에서 독립적으로 애플리케이션을 실행할 수 있도록 컨테이너를 만들고 관리하는 것을 도와주는 도구임. hyperledger fabric을 도커에 올려서 실행하면 편리한 운영이 가능하기에 사용하기로 하였다.

- 환경

환경	역할
AWS	웹페이지 로그인 DB를 구현할 서버. 로그인 DB는 Mysql을 사용할 예정
html5	REC거래 웹페이지로 크롬 웹페이지에 맞추어서 개발할 예정

- 데이터 베이스

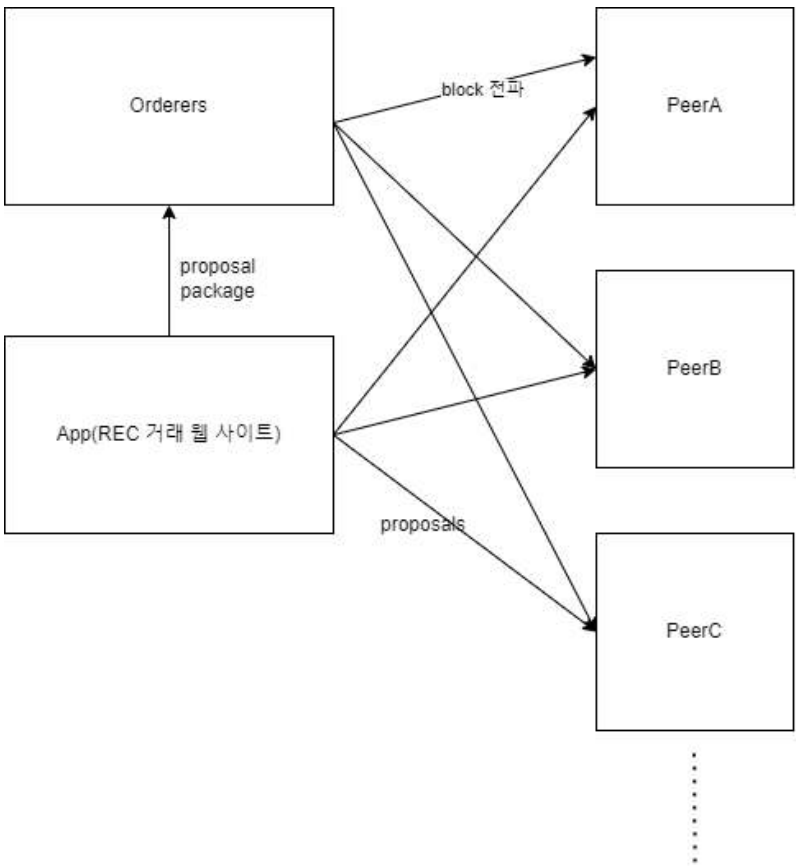
환경	역할
couch DB	hyperledger fabric에서 사용하는 db로 level DB보다 더 복잡한 query문을 사용할 수 있다는 장점이 있음
Mysql	oracle에서 개발한 sql언어로 사용 경험도 있고 무료라서 사용하기로 함

5.2 사용 기술

● hyperledger fabric

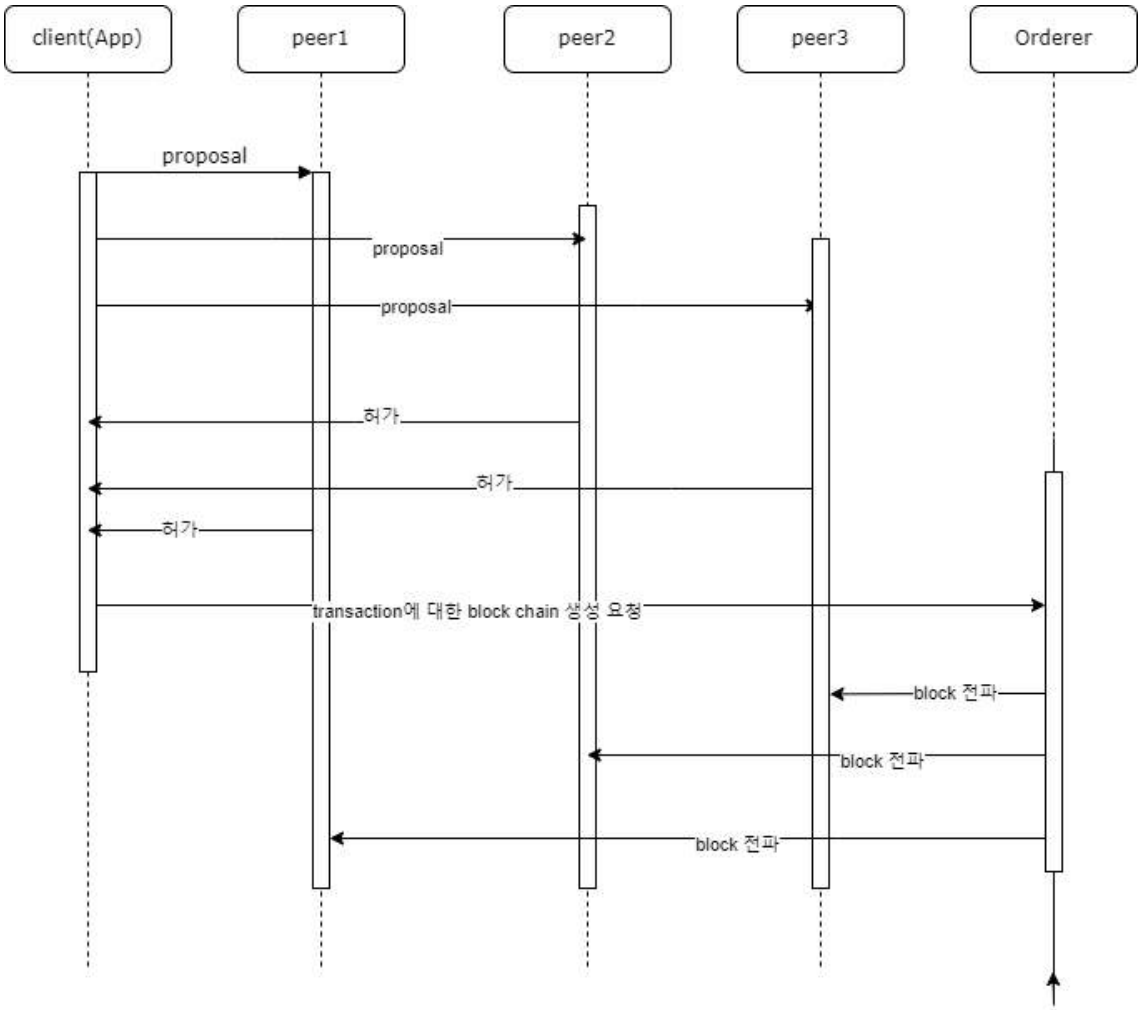
하이퍼레저는 리눅스 재단(Linux Foundation)이 주도하는 엔터프라이즈용 블록체인 기술 개발을 위한 오픈소스 프로젝트이다. 허가받은 사용자만 참여할 수 있는 허가형 블록체인으로서, 프라이빗 블록체인의 일종이다. 해당 플랫폼은 회원가입을 해서 허가된 사용자만 안전하게 사용하는 것이 목적이므로 프라이빗 블록체인에 매우 적합하여 하이퍼래저 패브릭을 사용한다.

hyperledger fabric 구동 방식



App(client)에서 거래를 만들어 낸다. 거래를 peer들에게 보내서 simulating(합당한 거래인지 확인하는 과정)한다. peer들에게 RAFT 합의 알고리즘을 통해서 거래서 합당한 것인지 평가를 받게 되고 합당하다고 판단 되면 orderer가 합당한 거래들을 차곡차곡 모아서 블록체인을 만들고 이 블록체인을 peer들에게 뿌리게 된다. peer들은 blockchain을 연결하고 world state를 update 한다.

sequence diagram으로 나타내면 다음과 같다.

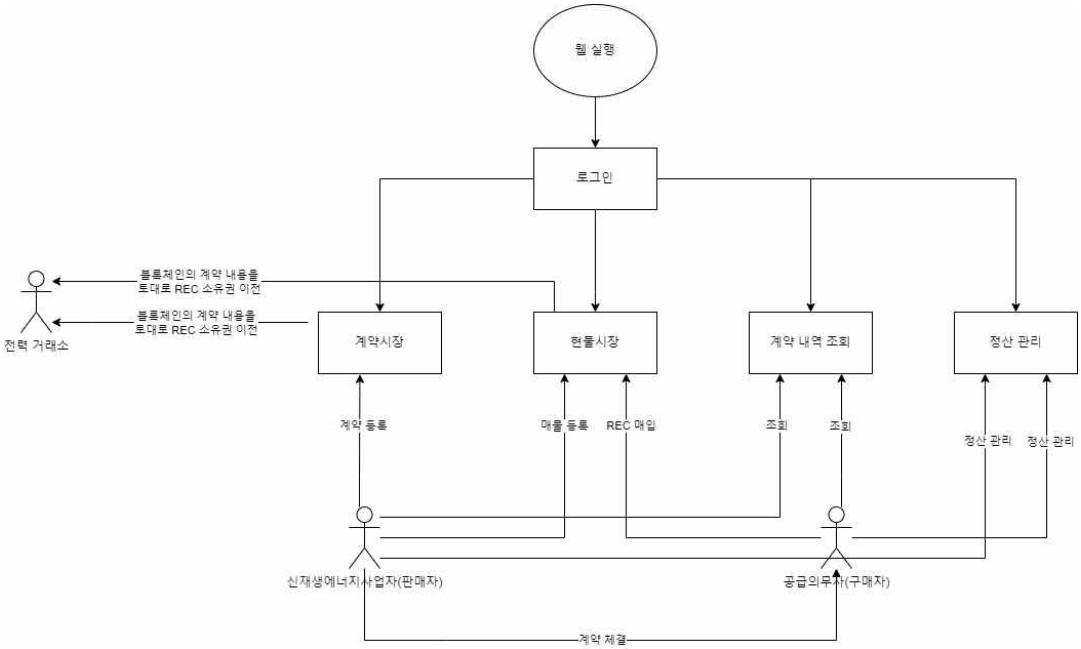


● 웹 개발

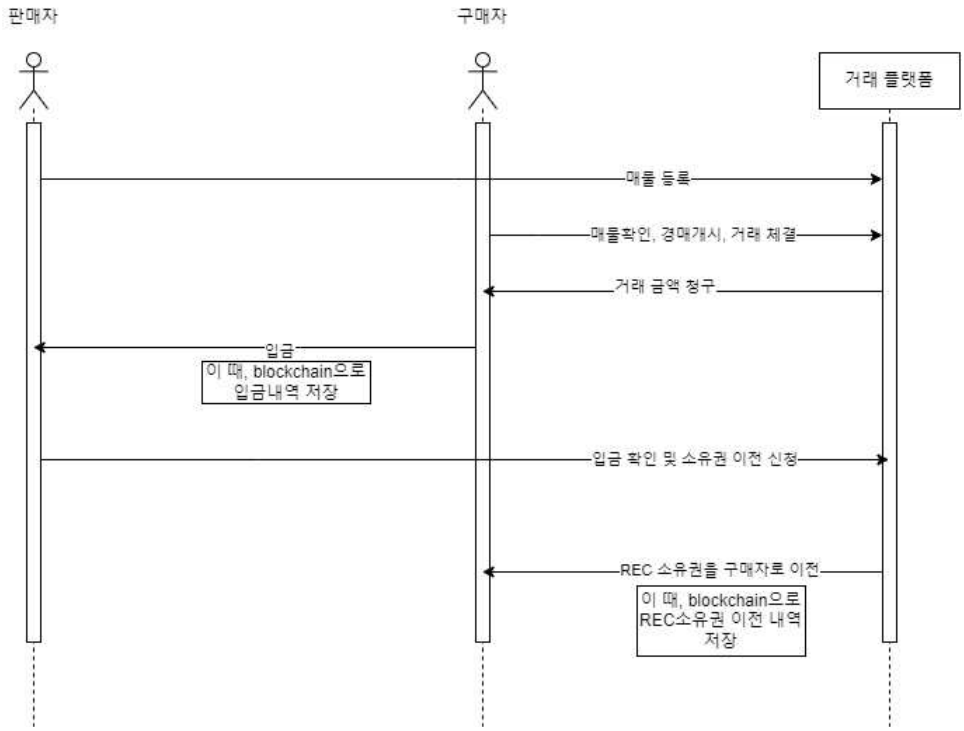
거래소는 typescript 와 react로 구현한다. 컴파일 단계에서 에러를 잡아낼 수 있다는 점에서 javascript에 비해서 더 높은 생산성을 가질 수 있다는 점에서 typescript를 웹 개발 언어로 채택하였고 좋은 성능을 위해 react를 사용하기로 하였다. 또한 JWT를 도입하여 로그인에서 보안성을 확보할 예정이다.

5.3 프로세스

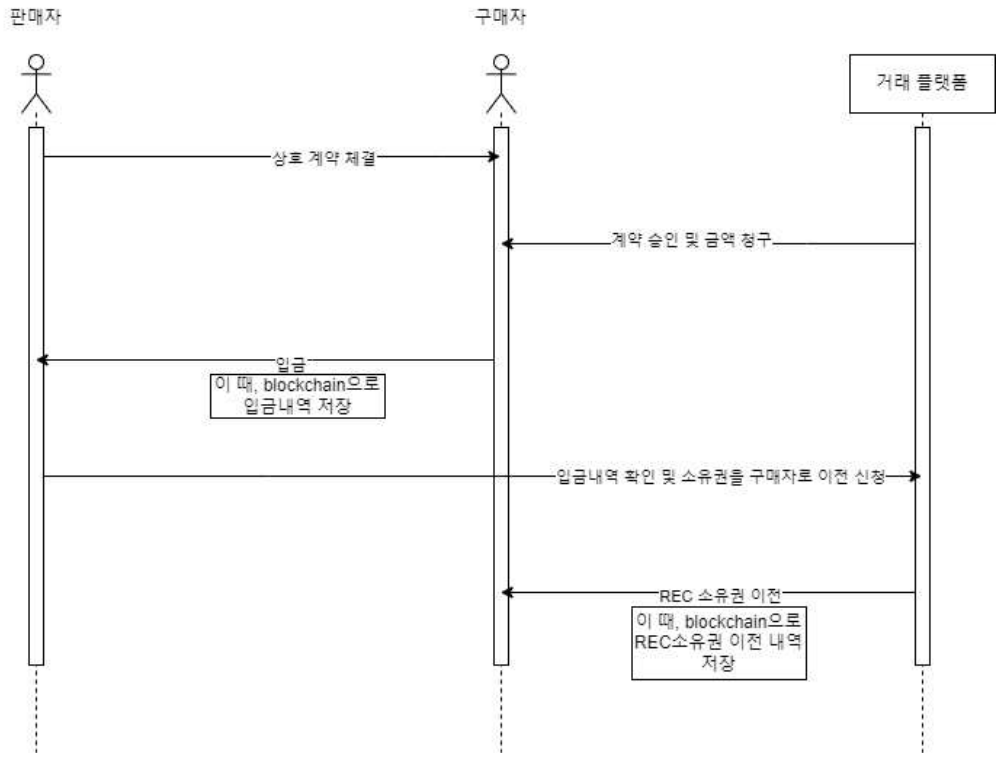
전체 구성도



계약시장 거래 과정 sequence diagram



현물시장 거래 과정 sequence diagram



6. 개발 일정 및 역할 분담

6.1 개발 일정

업무	6월				7월				8월				9월	
	1주	2주	3주	4주	1주	2주	3주	4주	1주	2주	3주	4주	1주	2주
기본 지식 공부														
데이터 서버 구축 및 플랫폼 구성														
하이퍼레저 패브릭 개발														
플랫폼 개발														
서버 개발														
테스트 및 디버깅														
최종 발표/ 보고서 준비														

6.2 역할 분담

차민준: - html, css, react 등을 이용한 웹페이지 디자인 및 기능 구현
- JWT를 활용한 안전한 로그인 시스템 개발
- 사용자 친화적인 user interface를 적용하여 front-end 개발

오재석: - Node.js와 typescript를 사용한 서버 개발
- AWS의 EC2, RDS 서비스를 사용해 서버 환경 구축
- back-end, front-end 전반적인 개발 assist

김재현: - 블록체인 서버 구현
- GO를 이용하여 블록체인 모듈 구현
- smart contract 개발

그림1, 그림2) The data transparency of RPS using blockchain method- [https://doi.org/10.9708/jksci.2020.25.03.081]